UML CLASS DIAGRAM – Efrén Pérez Marrero 1ºDAW

ENTITIES:

1. ***User***

Features (private):

- userId: String

- username: String

- password: String

Methods (public):

+ getDetails(): User

+ validateCredentials(username: String, password: String): Boolean

2. ***AuthenticationService***

Features (private):

- currentUser: User

Methods (public):

+ login(username: String, password: String): Boolean

+ logout(): void

+ signin(newUser: User): Boolean

3. ***SessionManager***

Features (private):

- activeSessions: Map<String, User> (almacenar sesiones activas)

Methods (public):

+ createSession(userId: String): void

+ destroySession(userId: String): void

+ isSessionActive(userId: String): Boolean

4. ***Database***

Features (private):

- users: List<User> (almacenar usuarios registrados)

Methods (public):

+ addUser(newUser: User): Boolean

+ findUserByUsername(username: String): User

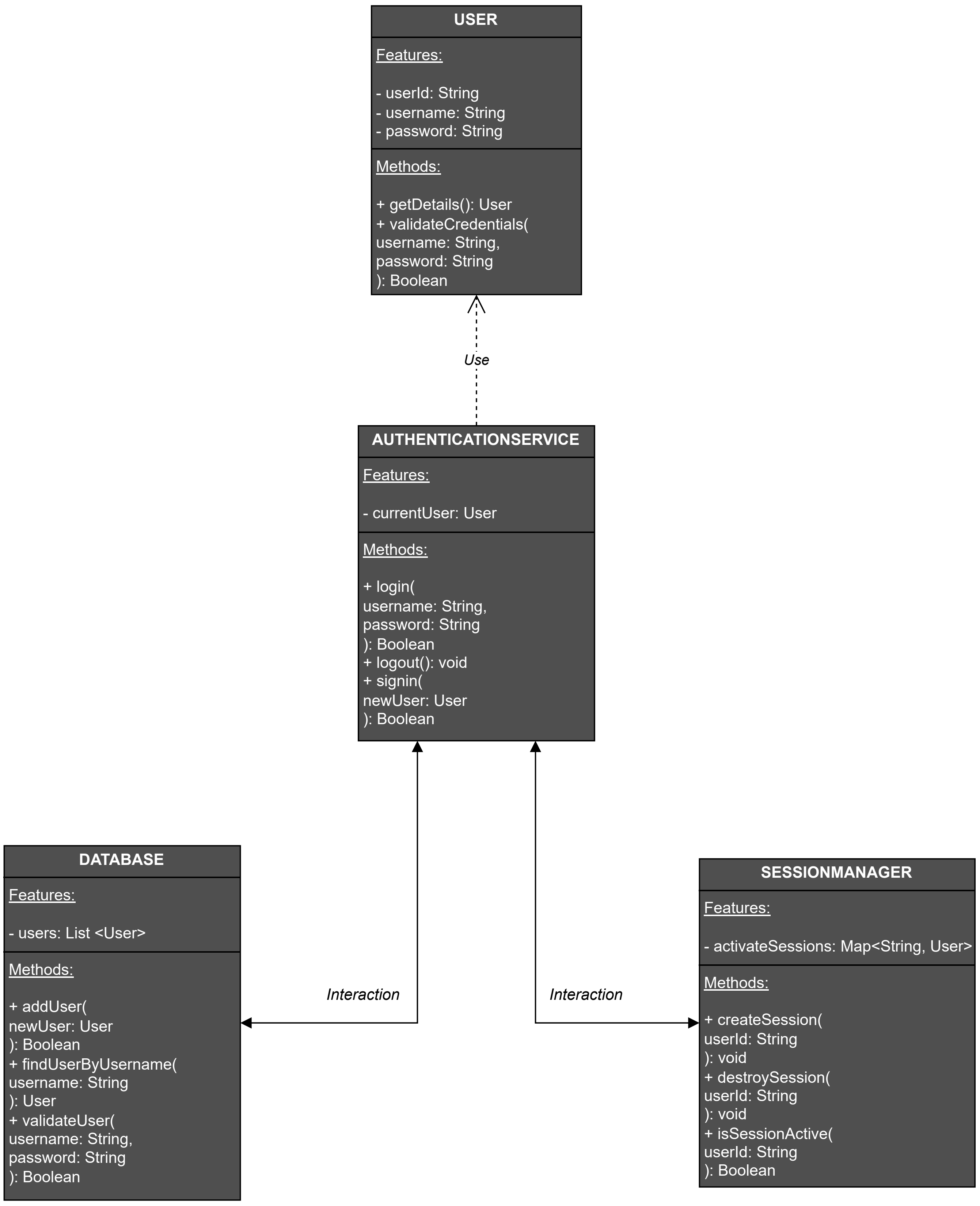
+ validateUser(username: String, password: String): Boolean

RELATIONS:

User is used by **AuthenticationService.**

AuthenticationService interacts with **SessionManager** to manage sessions.

AuthenticationService interacts with **Database** to find or validate users.


USER
Features:
- userId: String
- username: String
- password: String
Methods:
+ getDetails(): User
+ validateCredentials(
username: String,
password: String
): Boolean
Use
AUTHENTICATIONSERVICE
Features:
- currentUser: User
Methods:
+ login(
username: String,
password: String
): Boolean
+ logout(): void
+ signin(
newUser: User
): Boolean
Interaction
Interaction
DATABASE
Features:
- users: List <User>
Methods:
+ addUser(
newUser: User
): Boolean
+ findUserByUsername(
username: String
): User
+ validateUser(
username: String,
password: String
): Boolean
SESSIONMANAGER
Features:
- activateSessions: Map<String, User>
Methods:
+ createSession(
userId: String
): void
+ destroySession(
userId: String
): void
+ isSessionActive(
userId: String
): Boolean

UML Activity Diagram – Efrén Pérez Marrero 1ºDAW

UML activity diagram:

1.- User interacts with AuthenticationService.

2.- AuthenticationService calls Database to validate or find a user by username.

3.- If the user is found and validated, AuthenticationService interacts with SessionManager to create or destroy sessions based on login/logout actions.

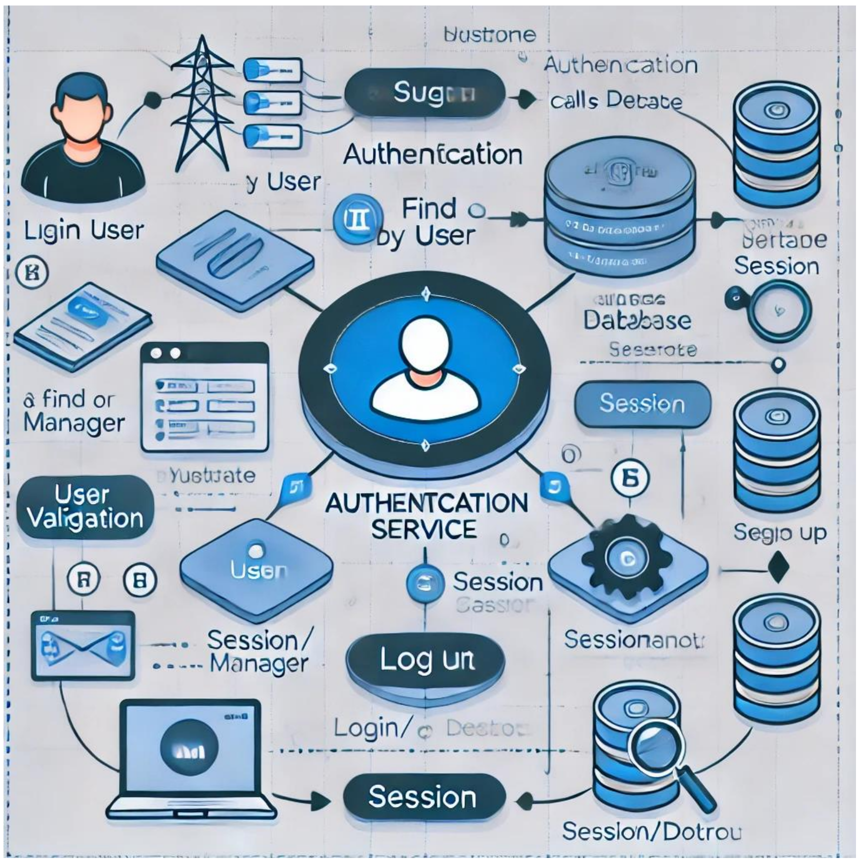
Flow System/Diagram:

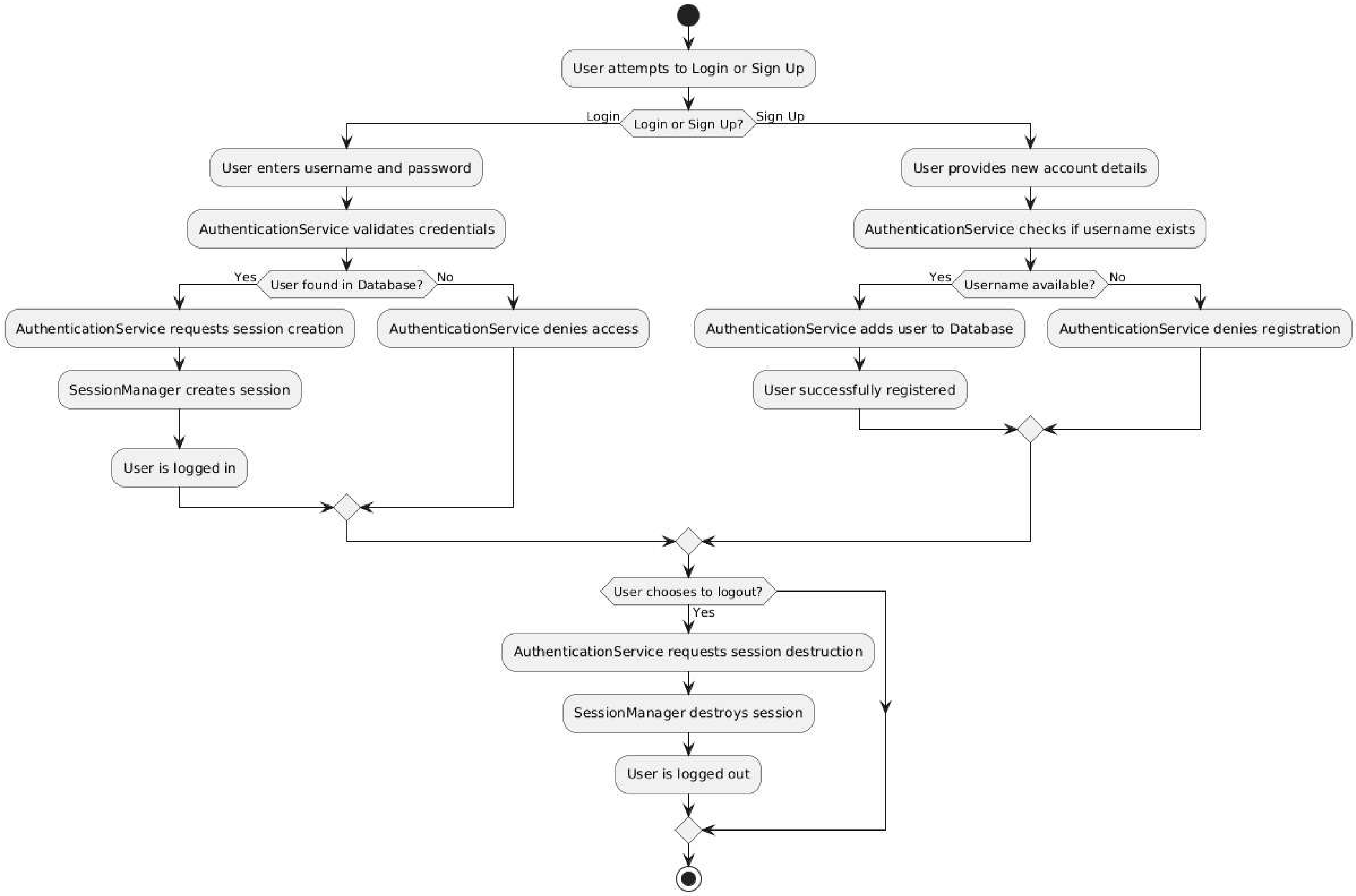
1.- The flow starts when the User tries to login or sign up.

2.- Then, AuthenticationService performs user validation.

3.- Based on the outcome, a session is created or destroyed.

4.- Unforgettable: It is necessary to include decision points for user validation, session creation, and session check.




User attempts to Login or Sign Up
Login
Login or Sign Up?
Sign Up
User enters username and password
AuthenticationService validates credentials
Yes
User found in Database?
No
AuthenticationService requests session creation
AuthenticationService denies access
SessionManager creates session
User is logged in
User provides new account details
AuthenticationService checks if username exists
Yes
Username available?
No
AuthenticationService adds user to Database
AuthenticationService denies registration
User successfully registered
User chooses to logout?
Yes
AuthenticationService requests session destruction
SessionManager destroys session
User is logged out